Videojuegos

Curso de Diseño y Programación

Nº 17 5,99 euros



*IA de los **Ovnis** y **Voladores***

***Texturizar** el terreno*

*Blitz3D y las **DirectShow***











Curso práctico de Videojuegos

AUTOR DE LA OBRA
Marcos Medina
DIRECCIÓN EDITORIAL
Eduardo Toribio
etoribio@iberprensa.com
COORDINACIÓN EDITORIAL
Eva-Margarita García
eva@iberprensa.com
DISEÑO Y MAQUETACIÓN
Antonio Gª Tomé
PRODUCCIÓN
Marisa Cogorro
SUSCRIPCIONES
Tel: 91 628 02 03
Fax: 91 628 09 35
suscripciones@iberprensa.com
FILMACIÓN: Fotpreim Duvial
IMPRESIÓN: Gráficas Don Bosco
DUPLICACIÓN CD.ROM: M.P.O.
DISTRIBUCIÓN
S.G.E.L.
Avda. Valdelaparra 29 (Pol. Ind.)
28108 Alcobendas (Madrid)
Tel.: 91 657 69 00

EDITA: Iberprensa
www.iberprensa.com



CONSEJERO
Carlos Peropadre

REDACCIÓN, PUBLICIDAD Y ADMINISTRACIÓN
C/ del Río Ter, 7 (Pol. Ind. "El Nogal")
28110 Algete (Madrid)
Tel.: 91 628 02 03
Fax: 91 628 09 35
(Añada 34 si llama desde fuera de España.)

DEPÓSITO LEGAL: M-35934-2002
ISBN: **Coleccionable:** 84 932417 2 5
Tomo 2: 84 932417 4 1
Obra Completa: 84 932417 5 X

NOTA IMPORTANTE:
Algunos programas incluidos en los CD de "Programación y Diseño de Videojuegos" son versiones completas, pero en otros casos se trata de versiones demo o trial, versiones de evaluación que Iberprensa quiere ofrecer a nuestros lectores. No se trata en ningún caso de las versiones comerciales de los programas, y las hemos incluido para dar al lector la oportunidad de conocer y probar esos programas y que así pueda decidir posteriormente si desea o no adquirir las versiones comerciales de cada uno.

Aprende divirtiéndote

Bienvenidos de nuevo a **Programación y Diseño de Videojuegos**, la primera obra coleccionable cuyo objetivo es formar al alumno en las principales técnicas relacionadas en el desarrollo completo de un videojuego.

A lo largo de la obra el lector está aprendiendo programación a nivel general y a nivel específico con ciertas herramientas y lenguajes, aprendiendo a trabajar con aplicaciones de retoque de imagen y también de diseño 3D y animación. Estamos descubriendo las aplicaciones profesionales más importantes de audio y conociendo la historia de lo que se denomina "la industria del videojuego", los últimos 20 años, los juegos que marcaron un avance, sus creadores y en general la evolución del videojuego.

Pero además, esta obra tiene un segundo objetivo, desarrollar y potenciar la creatividad del lector, nosotros a lo largo de las diferentes entregas pondremos las bases y tú pondrás tu ingenio, tu creatividad y tu capacidad de mejorar.

Nos encontramos a mitad de camino del viaje de 20 semanas que os proponemos, viaje articulado en 400 páginas y 20 CD-ROMs cuya finalidad es proporcionar las bases mínimas para después cada uno continuar su camino.

Recuerda que para alcanzar el éxito necesitas cumplir tres condiciones: que te gusten los juegos, poseer cierta dosis de creatividad y finalmente capacidad de estudio.

Una la cumples seguro.

sumario

321 Zona de desarrollo
Siguiendo con el desarrollo del comportamiento de los sujetos que participan en el juego pasamos a implementar la IA de los OVNIS y Voladores.

325 Zona de gráficos
En esta entrega vamos a realizar dos texturas que cubrirán el terreno que fabricamos en el número anterior.

329 Zona de audio
Con esta entrega terminaremos la serie dedicada al secuenciador Anvil Studio con el que realizamos la música de nuestro juego.

331 Blitz 3D
En este número completaremos nuestro aprendizaje del aspecto multimedia de Blitz3D viendo las funciones de audio y las posibilidades de reproducción de vídeos a través de la utilización que este lenguaje tiene de las DirectShow.

335 Tutorial
Continuamos en esta entrega con el montaje del vídeo de presentación de nuestro juego.

337 Historia del videojuego
Seguimos analizando la evolución de los simuladores, pero esta vez le toca el turno a los simuladores deportivos.

339 Cuestionario
Cada semana un pequeño test de autoevaluación, en el próximo número encontrarás las respuestas.

340 Contenido CD-ROM
Páginas dedicadas a la instalación y descripción del software que se adjunta con cada coleccionable.

17

PARA ENCUADERNAR LA OBRA:

- ***Tapas del volumen 1 disponibles en Iberprensa. Pedidos por teléfono: 916280203.***
- ***Tapas del volumen 2 ya a la venta en quioscos.***
- ***Los suscriptores recibirán las tapas en su domicilio sin cargo alguno como obsequio de Iberprensa.***

SERVICIO TÉCNICO:
Para consultas, dudas técnicas y reclamaciones Iberprensa ofrece la siguiente dirección de correo electrónico: games@iberprensa.com

PETICIÓN DE NÚMEROS ATRASADOS:
El envío de números sueltos o atrasados se realizará contra reembolso del precio de venta al público más el coste de los gastos de envío. Pueden ser solicitados en el teléfono de atención al cliente 91 628 02 03

Inteligencia artificial (II) OVNIS y voladores

Siguiendo con el desarrollo del comportamiento de los sujetos que participan en el juego pasamos a implementar la IA de los OVNIS y Voladores.

A diferencia de los demás animales y plantas, estos elementos tienen un factor que conlleva el desarrollo de una conducta diferente, el desplazamiento.

(■) COMPORTAMIENTO DE LOS OVNIS

Mientras las plantas carnívoras y gusanos permanecen en todo momento en la misma posición y su comportamiento se basa en cambios de animación, los OVNIS se trasladan continuamente con velocidad variable sobre el terreno en busca de nuestra bionave. Este hecho nos lleva a desarrollar una nueva técnica consistente en el manejo de variables de desplazamiento y de posición. Además, necesitamos construir un sistema que permita a los OVNIS buscar y atacar a nuestra unidad. Aparte de estas cualidades básicas para proporcionar un comportamiento más o menos razonable, es posible incluir algunos nuevos como:

-BUSCAR NUEVA POSICIÓN
-DESPLAZAR A LA NUEVA POSICIÓN
-HUIR AL RECIBIR UN IMPACTO
-DISPARAR SEGÚN RADIO DE ACCIÓN

1

Esquema de la conducta básica de los OVNIS: buscar, desplazar, huir, disparar.

seleccionar el tipo de armamento durante el ataque, cambiar de objetivo sobre la marcha, emprender la huida en caso de recibir un impacto o incluso buscar munición o energía según se agote. En las figuras 1 y 2 se muestran los esquemas del comportamiento.

Hemos querido implementar todo el sistema de comportamiento en un módulo aparte ("jugadores_CPU.bb") para facilitar y añadir nuevos comportamientos o depurar errores. En este módulo situamos las funciones para crear y actualizar los OVNIS.

Antes de entrar de lleno en el código de la IA, es necesario explicar cómo definimos y creamos los OVNIS.

(■) DEFINIENDO LOS OVNIS

Como es habitual, cada OVNI está definido por una estructura de datos con todos los parámetros que intervienen en su comportamiento en el módulo de definiciones. Como ocurre con los demás seres, esto nos proporciona un comportamiento independiente para cada uno de ellos:

```
Type tipo_jugadorCPU
  Field entidad,sombra_CPU,pivote
  Field x#,y#,z#,dxv#,dzv#,velocidad#
  Field vida
  Field municion, municion2,
  municion3, municion4
  Field tipo_armamento=1
  Field objetivo, distancia_objetivo
  Field sw_cambio
  Field contador%,contador_total%
End Type
```

Podemos observar que definimos los campos necesarios para identificar al OVNI y proporcionarle una sombra y un pivote. Los siguientes campos hacen referencia a su posición y a su capacidad de ataque:

-SELECCIONAR ARMA SEGÚN LA DISTANCIA AL OBJETIVO
-BUSCAR BONOS DE MUNICIÓN Y VIDA EN CASO DE TENERLAS AGOTADAS

2

Esquema de la conducta extendida de los OVNIS.

x, y, z almacenan la posición actual sobre el terreno; *dxv, dyv, dzv* las nuevas coordenadas de destino y *velocidad* almacena la velocidad de desplazamiento.

Con referencia a la capacidad de ataque tenemos *municion, municion2, municion3, municion4,* que hacen referencia a la cantidad de munición de que dispone y *tipo_armamento* al arma actual en uso. Para detectar al enemigo y marcar un punto de disparo usamos los campos *objetivo* y *distancia_objetivo*. Y para decidir los cambios de dirección en el desplazamiento disponemos de

Coordenadas de posición:
Coord. actual x, y, z
Nueva coord. dxv, dzv
Nueva velocidad velocidad
Control de decisión:
Ataque objetivo, distancia_objetivo
Cambio posición contador, contador_total, sw_cambio

3

Esquema de los campos de la estructura "tipo.jugadorCPU".

los campos *contador, contador_total y sw_cambio*.

(■) CREANDO LOS OVNIS

Al margen de esta estructura, debemos utilizar una matriz de 5 elementos para la creación aleatoria de cinco OVNIS diferentes (Fig. 4 y 5):

```
Dim bionaveCPU(4),
textura_bionaveCPU(4)
```

Como siempre, los modelos son cargados en la función "crear_modelos()" del módulo "funcpantaudio.bb":

```
For n=0 To 4
  bionaveCPU(n)=LoadMesh
  ("c:\zone of fighters\modelos\
  bionaveCPU.3ds")
  EntityTexture bionaveCPU(n),
  textura_bionaveCPU(n)
  ScaleMesh bionaveCPU(n),7,7,7
  EntityRadius bionaveCPU(n),70
Next
```

Volviendo al módulo "jugadores_CPU.bb" desarrollamos una función para crear tantos OVNIS como indique la variable "numero_jugadoresCPU". Se trata de "crear_jugadores_CPU()" (Ver código). En primer lugar, asignamos el número de luchadores al contador *OVNIS* en juego "contador_jugadorCPU", el cual se utiliza para saber si hemos ganado o no la partida. Para diferenciar un OVNI de otro utilizamos una estructura "Select..Case" para asignar, a través de un número aleatorio, cualquiera de los cinco modelos diferentes cargados anteriormente. Seguidamente, asignamos el tipo de entidad para el mapa de colisiones, creamos un pivote para marcar el origen de los disparos y procedemos a dar todos los valores al resto de los campos de la estructura. Destacamos la asignación de la última posición de la bionave (variables globales X y Z, correspondiente a sus coordenadas) a las variables de destino "jugadorCPU\dxv" y "jugadorCPU\dzv".



4

Imagen de los cinco tipos de OVNIS que se crean durante la partida.

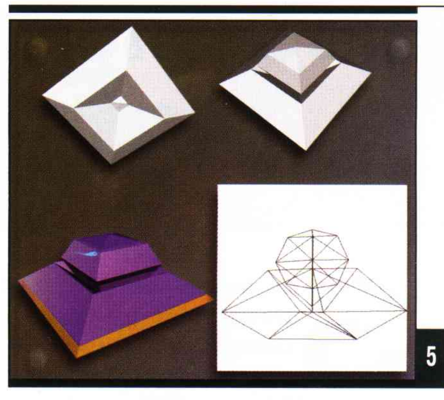
(■) IMPLEMENTANDO LA IA

Ha llegado el momento de actualizar la situación de cada OVNI en el terreno de juego desarrollando su comportamiento. Para ello, utilizamos la función "actualiza_jugadores_CPU()" para incluir el bucle que recorre la estructura de cada OVNI y en donde modificaremos todos sus estados. Describiendo básicamente esta función, comenzamos controlando la vida del OVNI y si ha sido alcanzado por algún disparo (tanto de la bionave como de otros OVNIS) (Ver código). En esta última comprobación incorporamos las sentencias para que el OVNI emprenda la huida asignando nuevas coordenadas de destino:

```
If EntityCollided(jugadorCPU\
entidad,ENTIDAD_DISPARO)
  ...
  jugadorCPU\dxv=Rnd(1000,13000)
  jugadorCPU\dzv=Rnd(1000,13000)
EndIf
```

Seguidamente, empezamos a comentar el código que conforma todo el comportamiento de los OVNIS (Fig. 6).

Para controlar la duración de las acciones vamos a usar el mismo sistema utilizado con las plantas carnívoras y gusanos; es decir, un contador ("jugadorCPU\contador" y "jugadorCPU\contador_total") que se decrementa en cada actualización del juego y que, según su valor, determina una acción u otra:



5

Modelo de los OVNIS. Para crearlos usamos una matriz de 5 elementos.

```
jugadorCPU\contador%=
jugadorCPU\contador-1
```

Así, si el contador llega a la mitad de su valor total y el OVNI no ha sufrido ningún cambio ("jugadorCPU\sw_cambio = 0") le asignamos otra dirección aleatoria de destino:

```
If jugadorCPU\contador<
jugadorCPU\contador_total/2
And jugadorCPU\sw_cambio=0
  jugadorCPU\dxv=Rnd(1000,13000)
  jugadorCPU\dzv=Rnd(1000,13000)
  jugadorCPU\sw_cambio=1
EndIf
```

En caso de que el contador llegue a cero, le asignamos un nuevo valor y dirigimos el OVNI hacia la última posición de la bionave con una nueva velocidad:

```
If jugadorCPU\contador=0
  jugadorCPU\contador=
  Rnd(800,2000): jugadorCPU\
```

POR CADA OVNI

- DECREMENTAR CONTADOR CONTROL AVANCES
- SI CONTADOR AVANCES / 2 ENTONCES CAMBIAR DIRECCION DESTINO ACTUAL
- SI CONTADOR AVANCES = 0 ENTONCES DIRECCION DESTINO = POSICION BIONAVE
- DESPLAZAR OVNI
- POSICIONAR OVNI
- AJUSTAR Y CALCULAR DISTANCIA A OBJETIVOS
- DISPARAR TIPO DE ARMA SEGÚN DISTANCIA
- CONTROL DE IMPACTOS CON BONOS
- BUSCAR MUNICIÓN O VIDA SI ESCASEA

6

Pseudocódigo de la IA de los OVNIS, en el que vemos qué acciones son capaces de realizar.

```
 contador_total=jugadorCPU\contador
 If camuflaje_bionave=0
 jugadorCPU\dxv=x :
 jugadorCPU\dzv=z
 jugadorCPU\velocidad=Rnd(4,8)
 jugadorCPU\sw_cambio=0
EndIf
```

Después de fijar las direcciones, modificamos sus variables de coordenadas según la velocidad contenida en el campo "jugadorCPU\velocidad". Para conseguir desplazarlo hacia las coordenadas de destino, marcadas por "jugadorCPU\dxv" y "jugadorCPU\dzv", comparamos su posición actual ("jugadorCPU\x" y "jugadorCPU\z") con estos campos. En caso de ser menor la incrementamos y en caso contrario la aumentamos (Ver Código 1).

Una vez calculadas las nuevas coordenadas, posicionamos el OVNI y apuntamos al objetivo marcado para llevar a cabo con éxito cada disparo:

```
PointEntity (jugadorCPU\pivote,
jugadorCPU\objetivo)
```

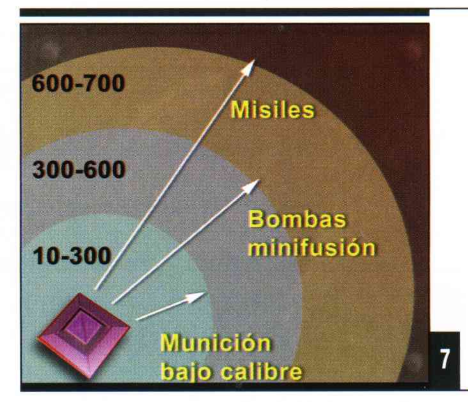
La asignación de valores al campo "jugadorCPU\objetivo" la explicamos a continuación.

El ataque de los ovnis se realiza siempre que la bionave no tiene el camuflaje activado ("camuflaje_bionave = 0"). Si esto ocurre, los OVNIS pueden "verla". Por lo tanto, asignamos la entidad "bionave" como objetivo ("jugadorCPU\objetivo") y calculamos su distancia para saber si es positivo o no atacar (Ver Código 2).

En caso de que sea imposible detectarla porque tiene el camuflaje activado, asignamos como objetivo de ataque otro OVNI. Después de calcular la distancia y de tener el objetivo marcado, pasamos al proceso de disparar. Asignamos un valor aleatorio a la variable "d" para determinar un promedio de disparos variable. Además, solo se dispara si la distancia al objetivo es inferior a 700 puntos. Si no hiciéramos esto, los OVNIS estarían continuamente disparando:

```
d=Rand(1,dificultad)
If jugadorCPU\
distancia_objetivo<700 And d=1
 ...
EndIf
```

La variable "dificultad" nos sirve para determinar ese promedio de disparos. Se dispara más si esta variable tuviera un valor pequeño, porque hay más posibilidades de que "d" valga 1, por ejemplo, entre 1 hasta 10 que entre 1 a 30. El siguiente paso es determinar qué tipo de armamento utilizar para el disparo. No sería lógico utilizar bombas de minifusión con un alcance de 300 puntos estando en una distancia de 600 (Fig 7).



Dependiendo de la distancia al objetivo se selecciona diferente armamento.

Por lo tanto, aprovechando que sabemos la distancia al objetivo por "jugadorCPU\distancia_objetivo", hacemos las comprobaciones pertinentes y en cada una asignamos la munición adecuada (Ver Código 3).

Una vez que ya sabemos qué arma utilizar pasamos a realizar el disparo según "jugadorCPU\tipo_armamento". Así que entramos en una estructura "Select.. Case" para disparar, siempre que tengamos munición (Ver Código 4).

El traslado del pivote es necesario para asignar de dónde saldrá cada disparo del OVNI y, sobre todo, a qué altura del terreno.

Código 1. Comparamos la posición del OVNI

```
If jugadorCPU\x#<=jugadorCPU\dxv#
  jugadorCPU\x=jugadorCPU\x+jugadorCPU\velocidad#
EndIf
If jugadorCPU\x>=jugadorCPU\dxv#
  jugadorCPU\x=jugadorCPU\x-jugadorCPU\velocidad#
EndIf
If jugadorCPU\z#<=jugadorCPU\dzv#
  jugadorCPU\z=jugadorCPU\z+jugadorCPU\velocidad#
EndIf
If jugadorCPU\z>=jugadorCPU\dzv#
  jugadorCPU\z=jugadorCPU\z-jugadorCPU\velocidad#
EndIf
```

Código 2. Calculamos la distancia de la bionave

```
If camuflaje_bionave=0
   jugadorCPU\objetivo=bionave
   jugadorCPU\distancia_objetivo#=EntityDistance (jugadorCPU\entidad,bionave)
Else
   jugadorCPU\objetivo=jugadorCPU\entidad
   jugadorCPU\distancia_objetivo#=EntityDistance (jugadorCPU\entidad,jugadorCPU\entidad)
Endif
```

Código 3. Asignamos la munición adecuada

```
If jugadorCPU\distancia_objetivo#>600 And jugadorCPU\distancia_objetivo#<700: jugadorCPU\tipo_armamento=3
 Else
 If jugadorCPU\distancia_objetivo#>300 And jugadorCPU\distancia_objetivo#<600: jugadorCPU\tipo_armamento=2
  Else
  If jugadorCPU\distancia_objetivo#>10 And jugadorCPU\distancia_objetivo#<300: jugadorCPU\tipo_armamento=1
   Else
    jugadorCPU\tipo_armamento=Rand(1,4)
  EndIf
 EndIf
EndIf
```

Ya solo nos queda implementar la inter-actuación de los OVNIS con los bonos. Tener en cuenta este aspecto es importantísimo para lograr un equilibrio en el juego. Manteniéndolo, podremos lograr una mayor jugabilidad. Debemos dar a los OVNIS las mismas oportunidades que tenemos nosotros ya que también a ellos se les contempla el gasto de munición y de vida. Por lo tanto, si chocan con un bono de munición de bajo calibre se les sumará (Ver Código 5).

Para finalizar, podemos realizar una implementación más compleja de este último comportamiento, decidiendo qué bono buscar según las necesidades de munición. Simplemente, tenemos que preguntar la cantidad de munición disponible para cada arma. En caso de necesitar alguna, buscar el tipo de bono que corresponda a esa munición y asignar su posición a las coordenadas de destino:

```
If jugadorCPU\municion<30
 For bonus.tipo_bonus=Each tipo_bonus
  If bonus\tipo=0
  jugadorCPU\dxv=bonus\bx
  jugadorCPU\dzv=bonus\bz
  EndIf
 Next
EndIf
...
```

Código 4. Si tenemos munición, disparamos

```
Select jugadorCPU\tipo_armamento
 Case 1
  If jugadorCPU\municion>0
   TranslateEntity jugadorCPU\pivote,0,10,0

crear_disparo.disp(jugadorCPU\pivote,jugadorCPU\tipo_armamento,disparo1CPU)
   jugadorCPU\municion=jugadorCPU\municion-1
   TranslateEntity jugadorCPU\pivote,0,-30,0
  EndIf
 Case 2
 ...
End Select
```

Código 5. Los OVNIS pueden chocar

```
If EntityCollided(jugadorCPU\entidad,ENTIDAD_BONUS)
  For n=1 To CountCollisions(jugadorCPU\entidad)
   ent=CollisionEntity(jugadorCPU\entidad,n)
  Next
  For bonus.tipo_bonus=Each tipo_bonus
   If bonus\entidad=ent
     Select bonus\tipo
      Case 0: jugadorCPU\municion=jugadorCPUmunicion+50
      ...
     End Select
      FreeEntity bonus\entidad : Delete bonus:
      numero_bonus=numero_bonus-1
      Return
     EndIf
   Next
EndIf
```

Como hemos podido comprobar, programar un comportamiento determinado resulta muy interesante y divertido. En el tintero quedan muchas mejoras, por ejemplo, la posibilidad de esquivar disparos o buscar zonas altas para aumentar la defensa. Solo es cuestión de imaginación.

(■) COMPORTAMIENTO DE LOS VOLADORES

Los VOLADORES se implementan en el módulo "Enemigos.bb" y su comportamiento se establece en la función "actualizar_voladores()". Su actitud es exactamente igual que los OVNIS, solo limitada a que no pueden disparar. Se desplazan por el terreno buscando la posición de la bionave u otra nueva posición según el valor del contador. En el código se puede observar cómo el funcionamiento es el mismo.

En el próximo número...

... implementaremos todo el audio del juego.

Texturizando el terreno de juego

En esta entrega vamos a realizar dos texturas que cubrirán el terreno que fabricamos en el número anterior. Para ello, utilizaremos de nuevo Paint Shop Pro y Bryce. La idea principal que se pretende es aprender una técnica para realizar texturas para nuestros propios terrenos.

Como sabemos, para texturizar el terreno de combate necesitaremos al menos una textura. En nuestro caso, usaremos la técnica "blending" para mezclar dos. Una de ellas tendrá el tamaño del mapa de alturas cubriendo por completo el terreno y la segunda, más pequeña, estará tileada.

CREANDO LA PRIMERA TEXTURA

Como hemos comentado, la primera textura tiene el mismo tamaño que el mapa de alturas. Así que, para realizarla con mayor facilidad, podemos utilizarlo como referencia. Ejecutamos Paint Shop Pro y cargamos el mapa de alturas que dibujamos en la entrega anterior.

OBTENIENDO EL TONO PRINCIPAL

Para empezar vamos a cambiar el tono general de color utilizando el balance de color en *Color Balance* en las opciones *Adjust* del menú *Colors*. Apliquemos los valores siguientes en *Midtones*: 24, -100 y -60 para obtener un tono marrón. Observamos cómo el río sigue de color negro. Para rellenarlo, lo seleccionamos con la herramienta *Magic Wand* (varita mágica) con un valor de tolerancia (*Tolerance*) del 80%. Seguidamente, rellenamos con el mismo tono marrón que podemos obtener utilizando el cuenta-gotas (herramienta *Dooper*). Igualmente, rellenamos el marco blanco que forma los muros en el mapa de alturas. En estos momentos, tenemos una textura bastante uniforme (Fig. 1).

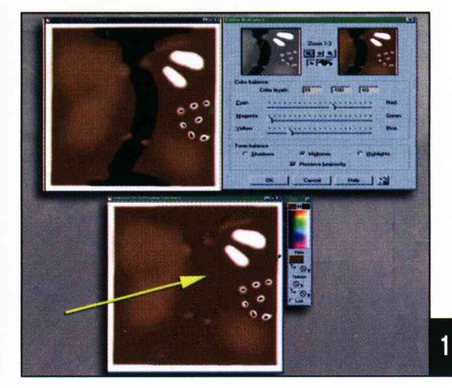
El siguiente proceso es dibujar poco a poco los diferentes matices de color sobre la textura que servirán para diferenciar cada parte del terreno de combate.

MATIZANDO LAS DIFERENTES ZONAS

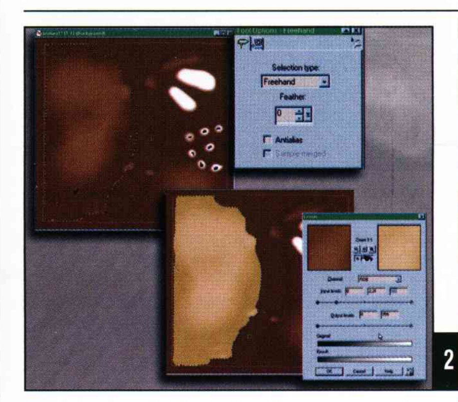
El trabajo de matización es realmente laborioso y es la parte más artística de la textura. Lo primero y más importante que debemos dibujar es la diferenciación básica de cada parte. Empezamos cambiando el tono del lado izquierdo del río, en donde ubicaremos las zonas verdes y la explanada para los almacenes y generadores. Elegimos la herramienta de selección en modo *freehand* y con mucho cuidado seleccionamos hasta el muro y bordeando la orilla del río.

Posteriormente, modificamos los niveles de color con la opción *Colors / Adjust / Levels* con los valores de entrada (Input levels) en 0, 2.31 y 255. Y con los valores de salida (Output levels) en 0 y 255 (Fig. 2).

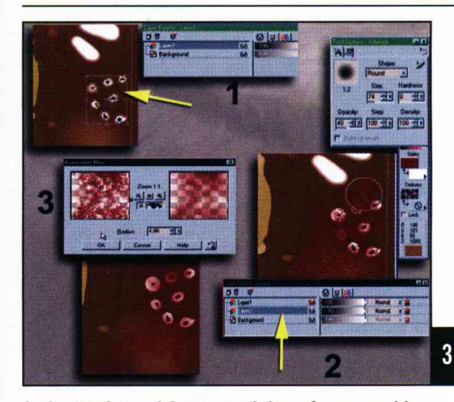
Seguimos por la zona de volcanes. En esta parte del terreno la tierra adquiere un tono rojizo. Para obtenerlo, vamos a pintar el terreno con la herramienta de spray (*Airbrush*) con un color rojo. Pero antes creamos una capa nueva con solo los volcanes para tenerlos ais-



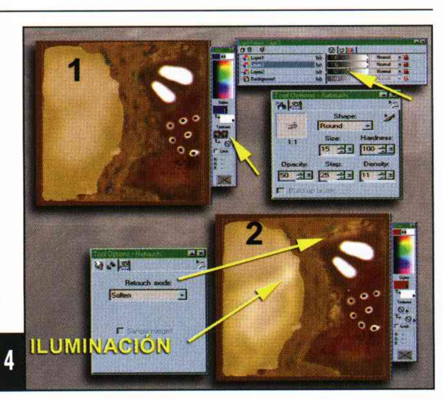
La primera de las texturas podemos obtenerla a partir del mapa de alturas.



Empezaremos dividiendo las diferentes zonas del terreno de juego.

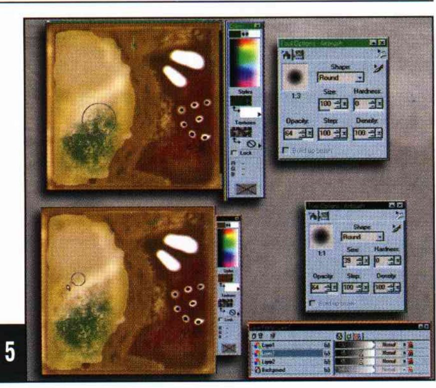


La herramienta del spray y el desenfoque son ideales para crear matices de color.



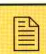
4

Mediante la herramienta de retoque Retouch en el modo Lighten RGB podemos obtener efectos de iluminación sobre la textura.



5

Es conveniente dibujar las diferentes zonas en distintas capas. Así, podemos aplicar efectos sin dañar el resto de la textura.



NOTA

Hay que recordar que para sumar selecciones tenemos que dejar pulsada la tecla "Shift" mientras seleccionamos.



NOTA

Es importante saber que la proporción de la textura en el juego con respecto al programa de dibujo se multiplica enormemente. Así, cualquier mancha o punto se vería aumentado considerablemente.

lados. Los seleccionamos con la herramienta *Magic Wand* (varita mágica) con una tolerancia de 110. Hacemos clic sobre el color blanco de uno de los volcanes, mantenemos pulsada la tecla "Shift" y hacemos clic en otro volcán. Así sucesivamente hasta tenerlos todos seleccionados. Seguidamente, creamos la capa con "Ctrl" + "C" y "Ctrl" + "L". Una vez creada la ocultamos y activamos de nuevo la capa de fondo (Fig. 3 / 1).

Elegimos el color blanco para la tinta del fondo en el panel *Colors* y pulsamos la tecla "suprimir" para borrar del fondo los siete volcanes seleccionados. Ahora que tenemos en una capa aparte los volcanes podemos pintar sin problemas el fondo con el spray. Este procedimiento nos ha evitado tener que jugar con las selecciones para evitar pintar los volcanes. Sin embargo, no pintaremos directamente en el fondo. Vamos a crear una nueva capa que utilizaremos para realizar este procedimiento, ya que posteriormente aplicaremos algunos efectos solo a esta zona. Para pintar con el spray seleccionamos la nueva capa, cambiamos su opacidad (en *opciones*) al 40 % y elegimos un tamaño del 75%. En *Textures* del panel *Colors* seleccionamos la textura *Terrain*, lo que dará un aspecto menos lineal (Fig. 3 / 2).

Después de pintar, aplicamos a la capa un efecto de desenfoque gaussiano en *Effects / Blur / Gaussian Blur* de 5.00 de radio. Con este procedimiento integramos los matices de color en el fondo (Fig. 3 / 3). Hagamos la misma operación para proporcionar a la misma zona algunos tonos diferentes de marrón para evitar un terreno extremadamente uniforme (Fig. 4 / 1).

Creamos una capa nueva y en ella pintamos con la herramienta *spray*. Además, en esta misma capa vamos a pintar los bordes del río para ocultar la línea divisora, siempre utilizando diferentes tonos de color. Lo importante a tener en cuenta es que hagamos divisiones con tonos diferentes para lograr separar las alturas. Por ejemplo, pintar los límites del río nos sirve para distinguirlo del resto del terreno más elevado, sobre todo cuando la cámara está en posición cenital. En este momento, tendremos un montón de manchas de diferente color a un lado del río. Para suavizar aún más la relación entre estas capas de matizados y el fondo vamos a cambiar su opacidad en un 65 %. El siguiente paso será suavizar todos los bordes de la capa del fondo marcados por los diferentes tonos de color con la herramienta de retoque *Retouch* en modo *Soften* (Fig. 4 / 2). Antes, ocultemos las demás capas para facilitar el trabajo. Siguiendo con el fondo, aprovecharemos para aplicar algunos retoques de iluminación como se muestra en la figura 4 / 2 con la misma herramienta anterior pero en el modo *Lighten RGB*. Esto nos ayudará a diferenciar las dos partes del río.

Seguimos ahora, con la mitad izquierda del río donde se sitúa la zona verde y la desértica. Para la primera, utilizamos de nuevo la herramienta *spray* pero con tonos de color verde. Seleccionamos un tamaño del *Shape* de 100 y hacemos clic un par de veces sobre la parte correspondiente en el terreno (Fig. 5).

Antes de pintar activamos y seleccionamos la última capa donde realizamos las anteriores matizaciones. No olvidemos que debemos utilizar la herramienta *Retouch* en modo *Soften* para desenfocar todos los espacios dejados por el spray.

Pasemos ahora a dibujar el tono para los volcanes. Ocultamos todas las demás capas, incluido el fondo, y nos quedamos solo con la capa de los volcanes. Nos acercamos con la herramienta de lupa y con el spray perfilamos el borde

del cráter con un tono granate y el resto con un tono marrón oscuro. Posteriormente, desenfocamos con la herramienta de retoque (*Retouch*). No olvidemos que los cráteres son puntos de color negro. Las dos montañas que forman el desfiladero las dejaremos en blanco pero las mancharemos un poco con tonos marrón.

Listo. Ya hemos terminado con la primera textura.

CREANDO LA SEGUNDA TEXTURA

La segunda textura se mezcla con la principal y se utiliza para proporcionar al terreno el detalle que necesita. En realidad es la que da el aspecto de tierra ya que, al no estar escalada al tamaño del mapa, su dibujo no se amplía, evitando la consiguiente pérdida de calidad. Aun así, la textura original tendrá el mismo tamaño, es decir, 512 x 512, simplemente, para que los detalles se aprecien mejor. Lo que ocurre es que durante su asignación al terreno se escalará a un tamaño inferior, con lo cual provoca un tileado para poder cubrirlo. Dado que la textura estará repetida, es decir, copias iguales colocadas una al lado de otra, necesitaremos dibujarla de tal forma que al unirlas los extremos coincidan.

FRABRICANDO LA TEXTURA BASE CON BRYCE

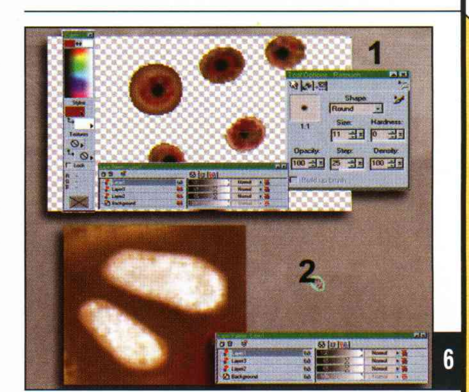
Para realizar la segunda textura vamos a aprovechar la técnica aprendida en el número 15 de esta sección para obtener texturas utilizando el programa Bryce. La textura obtenida nos servirá de base para realizar la definitiva. Ejecutemos Bryce. Empezamos cambiando el tamaño del documento a 512 x 512 en *File/Document Setup*. Creamos un cubo del menú *Create* y con la herramienta de desplazamiento del menú *Edit* movemos el cubo hasta situarlo a nuestra altura y lo escalamos hasta cubrir toda la vista.

Recordamos que el procedimiento que vamos a realizar consiste en texturizar el cubo y renderizar en el disco la imagen de la cara que cubre la vista. Entramos en el laboratorio de materiales y elegimos de la librería *Planes & Terrains* la textura *Desert Rock*. En las opciones de textura vamos a cambiar el *Bump Height* y *Ambience*. El primero con un valor de 4.00 y el segundo con valor máximo. Seguidamente, vamos a entrar en *Texture Source Editor* para cambiar los colores de la textura. Para entrar en esta sección tenemos que pulsar sobre la segunda esfera de la parte superior que se encuentra en el editor de la textura "A" (Ver Fig. 7 / 1).

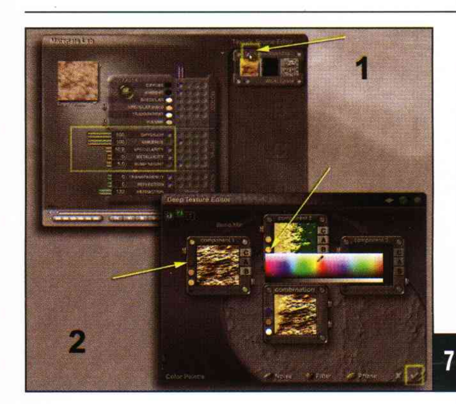
Esta nueva sección sirve para crear una textura a partir de la combinación de tres componentes en los que se combinan diferentes efectos de color y filtros de imagen con tres canales de salida cada uno: color, alfa y bump. En nuestro caso, la textura (*combination*) está formada por la combinación de los dos primeros componentes. Lo que nos interesa es cambiar el primer color marrón del primer componente a un tono más claro. Para ello, pulsamos sobre el primer círculo de color marrón y sin dejar de hacerlo elegimos un tono más claro de la paleta de colores. Hacemos lo mismo con el segundo color (naranja) del segundo componente (Fig. 7 / 2). Una vez que tenemos el cubo texturizado, movemos el sol hacia la parte inferior de la esfera para conseguir una textura más iluminada. Una vez realizado esto, renderizamos la imagen en el disco con la opción *File / Render to Disk*.

FABRICANDO LA SEGUNDA TEXTURA

Pasemos a Paint Shop Pro para realizar la textura definitiva. Cargamos nuestra textura base y creamos un nuevo documento de 512 x 512 en donde iremos formando la textura defini-



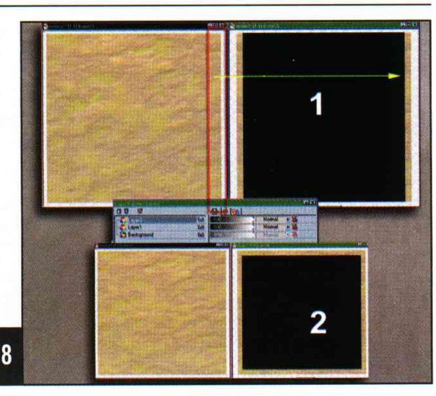
La escala del dibujo se ve multiplicada en la visualización del terreno. Cualquier punto en la textura corresponderá a una gran mancha en el visualizado final.



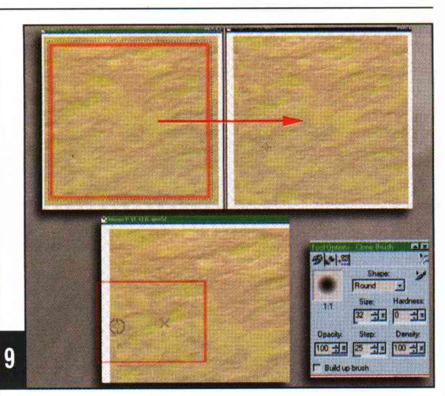
Bryce es ideal para obtener texturas base con numerosos detalles, ya que podemos modificarlas de múltiples maneras.

TRUCO

Debido a la diferencia de escala entre la textura dibujada y el resultado en el juego, el desenfoque es de vital importancia para evitar manchas de color demasiado definidas. La mejor manera de realizar un desenfoque controlado y selectivo es utilizando la herramienta de retoque *Retouch* en el modo *Soften*.



8 Debido a que la segunda textura estará tileada sobre el terreno es necesario hacer que los extremos, una vez unidos, coincidan entre sí.



9 La herramienta de clonado es ideal para tapar desperfectos o disimular las uniones entre trozos de textura.



10 Ejemplos del resultado final de nuestras texturas sobre el terreno.

tiva. Como comentábamos, la idea era que los extremos unidos de la textura coincidieran para poder utilizar un tileado. Así que seleccionamos un trozo más o menos ancho del extremo derecho de la textura base y creamos una nueva capa con "Ctrl" + "C" y "Ctrl" + "L". Seguidamente, pinchamos sobre esa capa y la trasladamos hacia el nuevo documento para colocarlo en la misma posición (extremo derecho). En el nuevo documento, copiamos el extremo adquirido con "Ctrl" + "C" y "Ctrl" + "L" para obtener el extremo izquierdo (Fig. 8 / 1).

Una vez creada la nueva capa, la seleccionamos, le hacemos un espejo horizontal con *Image / Mirror* ("Ctrl" + "M") y lo situamos en la parte izquierda del documento. A continuación, hacemos las mismas operaciones para los extremos superiores e inferiores (Fig. 8 / 2). En el documento de la textura base, no olvidemos borrar la capa del trozo de extremo que creamos. Para realizar un espejo vertical usamos la opción *Image / Flip*.

El problema ahora radica en que tenemos que cubrir la parte central del nuevo documento con el resto de la textura. Y evidentemente, los extremos no coincidirán. De todas formas, hagámoslo. Seleccionamos de la textura base el centro del documento menos los extremos que habíamos copiado y creamos una nueva capa. Llevemos la capa al segundo documento. Vemos que los extremos superior, inferior e izquierdo no coinciden con el resto de la textura (Fig. 9 / 1).

NOTA

Para ver el resultado de las texturas que hagamos podemos utilizar el visualizador de terrenos o el editor de "Zone of Fighters". Simplemente, tenemos que sustituir las texturas originales por las nuestras.

Una solución a este conflicto sería utilizando la herramienta de clonado . Una vez elegida, el cursor se transforma en una cruz rodeada por un círculo indicando el tamaño de la selección (puede ser modificada en el panel de opciones de la herramienta). El programa nos pide qué parte de la imagen queremos clonar. Para ello, pulsamos la tecla "Shift" izquierda mientras hacemos clic en la zona que queremos copiar. Al hacer clic se oirá un pitido que nos indica que la selección está hecha. Ahora, solo tenemos que dibujar en el lugar de destino. Observamos cómo una cruz permanece en el lugar de origen mientras dibujamos en el lugar de destino. Precisamente, este lugar corresponderá a la unión entre la capa del extremo y la capa que forma el resto de la textura (Fig. 9 / 2).

Cuando hayamos terminado unimos todas las capas para obtener una imagen única. Para realizarlo, hacemos clic sobre una de las capas con el botón derecho del ratón y elegimos la opción *Merge / Merge all (flatten)*.Ya podemos salvar la textura en formato PNG.

Ya solo nos queda matizar la textura para darle el aspecto que queramos. Para evitar que se note el tileado es conveniente no abusar de manchas de color muy definidas, etc. Tampoco conviene la utilización de desenfoque ya que perderíamos el propósito de esta textura: el detalle.

Utilizando las técnicas expuestas en estas últimas entregas será fácil la realización de cualquier terreno para nuestros juegos y en especial para "Zone of Fighters".

En el próximo número...

... dibujaremos los indicadores de pantalla y prepararemos las texturas de los bonos.

Nuestro primer tema musical con Anvil Studio (yVI)

Con esta entrega terminaremos la serie dedicada al secuenciador Anvil Studio con el que realizamos la música de nuestro juego.

Aprenderemos a crear y usar loops y además a desarrollar nuestros propios programas de efectos MIDI a través del "Performer".

LOOPS

La creación de loops nos permite insertar en los compases grupos de notas que se repiten; por ejemplo, para añadir arpegios. Vamos a desarrollar un pequeño loop (bucle). En primer lugar creamos una nueva canción con una sola pista "Instrument". Entramos en la sección *Compose* en el modo de edición *Piano Roll*. En la casilla del tamaño de la cuadrícula "Grid" elegimos 1/16 y en "Note", 1/8. En el primer compás dibujamos una nota al principio de cada parte. En C6, E, G y E de nuevo, para formar un pequeño arpegio de cuatro notas. Nos situamos en la línea de tiempo y seleccionamos las cuatro notas desplazando el ratón hacia la derecha. Para crear el loop elegimos la opción *Repeat selected notes in a Loop* del menú *Edit*. Aparece una ventana de diálogo en le que dejaremos las opciones por defecto (seleccionado *Just play loop once*. Tocar el loop una sola vez) (Ver Fig. 1).

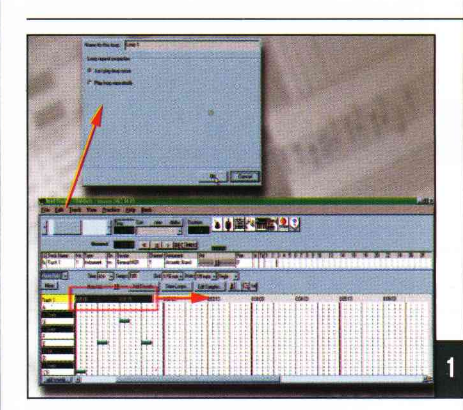
Al aceptar aparece debajo de la cuadrícula del "Track 1" otra cuadrícula perteneciente al nuevo loop "Loop 1". Podemos observar cómo en esta nueva cuadrícula aparecen las cuatro notas seleccionadas anteriormente. Además, en la cuadrícula "Track 1" también aparece una nueva división llamada "Loop 1" con una línea azul similar a una nota con la longitud de la selección. Esta línea significa que es un loop y que podemos desplazar a lo largo del tiempo como queramos. Podemos añadir más notas en el compás donde esté el loop ya que éste sonará independientemente del resto de notas. Si queremos que el loop tenga una duración mayor, debemos decirle al programa cuántas veces deseamos que se repita el loop. Para ello, abrimos la ventana de propiedades haciendo clic con el botón derecho del ratón sobre la línea azul que representa al loop. Pinchamos en la opción *Play loop repeatedly* y seleccionamos, por ejemplo, un 2 en la casilla *Number of times loop should play*. Al aceptar, observamos cómo la línea azul es el doble de larga y además al principio de la misma aparece una pequeña flechita curva indicando que el loop está repetido (Ver Fig. 2).

Aun formado el loop, podemos modificarlo. Para editarlo lo tenemos que hacer activo, así que pulsamos sobre su nombre "Loop 1" (en su cuadrícula) y se volverá de color amarillo. A partir de ahí, ya podemos editar las notas del loop de forma habitual.

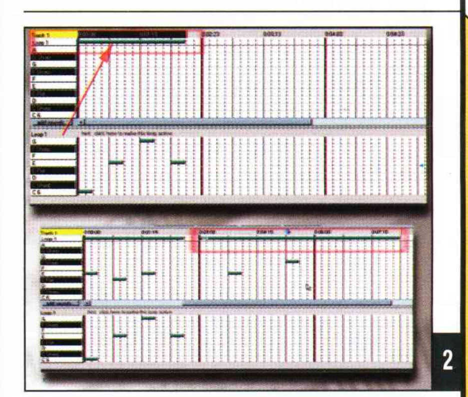
Puede ocurrir que tengamos más de un loop y que solo nos interese ver la cuadrícula de uno de ellos. Para elegir qué loops visualizar, pulsamos en el botón "Show Loops...". Abriremos la ventana *Select the loops you want you see* en donde podemos elegir qué loops ocultar o visualizar solo con hacer clic sobre su nombre (Ver Fig. 3).

EL PERFORMER

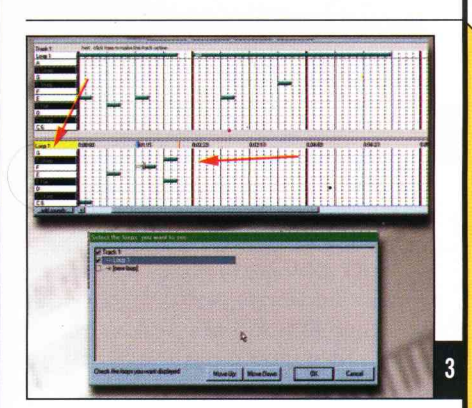
Una cualidad muy interesante que permite Anvil Studio es la posibilidad de crear pequeños programas MIDI por medio de la unión de iconos que representan


1

Procedimiento para crear un loop (repetición) de un arpegio de cuatro notas.


2

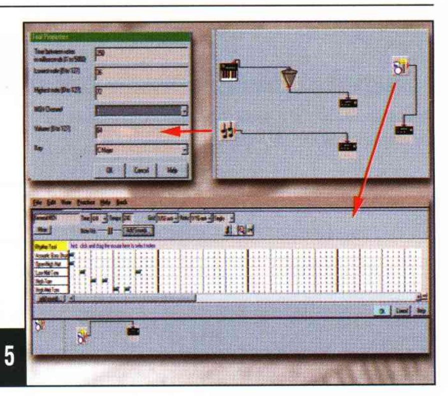
Podemos repetir un loop cuantas veces queramos y situarlo en cualquier lugar de la línea de tiempo de la canción.


3

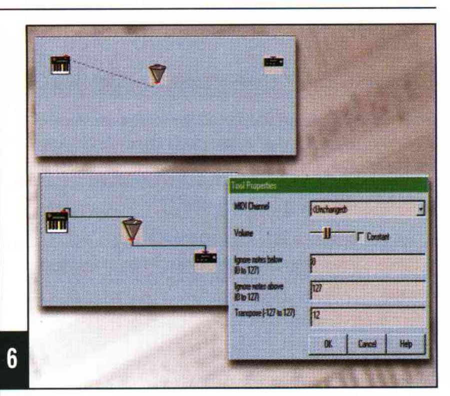
Es posible modificar cada nota de un loop creado con anterioridad.



Descripción de cada uno de los iconos de herramientas del Performer.



Cada herramienta tiene sus propias propiedades. Para acceder a ellas solo hay que pulsar con el botón derecho sobre el icono de la herramienta.



Procedimiento para crear un sencillo ejemplo en el que se utiliza un convertidor de notas para aumentar el tono en tiempo real de las notas tocadas.

diferentes herramientas sonoras. Para acceder al *Performer* elegimos la opción *Perfomer* en el menú *View*. En la figura 4 se describen las diferentes partes de esta sección.

La primera herramienta correspondiente al icono de un teclado representa el teclado MIDI o el virtual situado en la parte inferior. La herramienta del módulo de sonido (Sound Module) indica hacia dónde será desviada la reproducción de los eventos MIDI. La tercera herramienta es un convertidor de notas (Note Converter) que puede transformar en tiempo real el tono y el volumen de un grupo de notas. La siguiente herramienta es un generador de notas aleatorio (Random Note) también en tiempo real, con el que se pueden crear efectos sonoros muy interesantes. Podemos elegir el intervalo entre notas, seleccionar qué grupo de ellas sonará o incluso el volumen y el tono de cada una. La última herramienta de que disponemos es un reproductor de ritmos, los cuales creamos en una cuadrícula similar al modo *Piano Roll* (Ver Fig. 5).

CREANDO UN PROGRAMA

Para crear un nuevo programa solo tenemos que pulsar en el botón "New". En la zona de edición aparecerá la configuración básica; es decir, una herramienta de producción de notas (el teclado) y otra para reproducirlas (el módulo de sonido). Ejecutemos este programa base pulsando en el botón "Run". A continuación, el programa espera que se pulse alguna tecla del teclado conectado al ordenador o bien del virtual presente en pantalla. Este ejemplo básico de programa solo hace sonar las notas que toquemos. Por ejemplo, supongamos que además de esto queremos que las notas se transporten en tiempo real una octava más alta antes de sonar. Para ello, solo tenemos que insertar la herramienta *Note Converter* (convertidor de notas) entre las dos herramientas básicas anteriores. Para realizar este inserto hacemos clic sobre el tercer icono de la lista de herramientas. Seguidamente, hacemos clic entre el teclado y el módulo de sonido. Observamos cómo se coloca un icono del convertidor de notas. Ahora tenemos que romper la unión entre el icono del teclado y el módulo para unir el teclado al convertidor y éste al módulo. Esto significa que la salida del teclado (punta de flecha de color rojo) va hacia la entrada del convertidor (punta de flecha de color azul) y la salida de éste hacia la entrada del módulo de sonido. Para realizar esta operación, primero tenemos que cortar la unión entre los dos módulos. Nos situamos con el ratón sobre la entrada (flecha azul) del módulo de sonido. El puntero en forma de mano se convertirá en una cruz. Cuando esto ocurra hacemos clic provocando el corte en la unión. Para unir el teclado con el convertidor, nos situamos en la salida del primero hasta que el cursor con forma de mano abierta cambie a una mano cerrada. Sin dejar de pulsar el botón del ratón, nos desplazamos hacia la entrada del convertidor. Vemos cómo se va dibujando una línea discontinua detrás de la mano. Cuando lleguemos a la entrada soltamos el botón del ratón. Se dibujará una línea indicando que hemos conectado ambas herramientas. Para finalizar, debemos unir la salida del convertidor a la entrada del módulo de sonido.
Ejecutamos el programa pulsando en "Run" y tocamos en el teclado (Ver Fig. 6).

Podemos oír cómo las notas que tocamos suenan una octava más aguda.

Es posible salvar nuestro programa en disco para utilizarlo o modificarlo posteriormente con los botones "Load","Save" y "Save As".

Podemos realizar cuantas combinaciones queramos siempre con cierta coherencia.



En el próximo número...

... empezaremos con los trackers.

Audio y vídeo en Blitz 3D

En este número completaremos nuestro aprendizaje del aspecto multimedia de Blitz3D viendo las funciones de audio y las posibilidades de reproducción de vídeos a través de la utilización que este lenguaje tiene de las DirectShow.

AUDIO

Blitz3D permite cargar hasta cuatro tipos de formatos de audio diferentes en memoria: RAW, WAV, MP3 y OGG. Para ello, utilizaremos la instrucción "LoadSound":

```
Global Sonido
Sonido = LoadSound
("C:\disparo.wav")
```

REPRODUCCIÓN Y EDICIÓN BÁSICAS

Una vez que tenemos el sonido en memoria podemos utilizarlo de diferentes maneras. Lo más básico es reproducirlo. La forma más sencilla es utilizando "PlaySound":

```
Global Sonido
Sonido = LoadSound
("C:\disparo.wav")
Tocar_Sonido = PlaySound (Sonido)
```

Formatos de audio

WAV, MP3, OGG, RAW

Formatos de música

RAW, MOD, XM, S3M, MID, IT, WAV, MP2, MP3, OGG, ASF, WMA

1

Formatos de sonido y música soportados por Blitz3D.

Aunque podemos resumirlo de la siguiente manera:

```
Global Sonido = PlaySound
(LoadSound ("C:\disparo.wav"))
```

Es importante recordar que una vez que el sonido no es usado conviene eliminarlo de la memoria, por ejemplo, cuando tenemos varios niveles en un juego con diferentes sonidos. Para realizar esta operación usaremos "FreeSound":

```
FreeSound Variable del sonido
```

También podemos hacer un bucle sin fin en la reproducción del sonido utilizando la instrucción "LoopSound" antes de "PlaySound". Este sistema es ideal para reproducir fondos sonoros, por ejemplo, para simular el viento o la lluvia:

```
Global Sonido = LoadSound
("C:\viento.wav")
LoopSound Sonido
Sonar_Viento = PlaySound (Sonido)
```

Si queremos variar el volumen de reproducción, disponemos de la instrucción "SoundVolume":

```
SoundVolume Variable del
Sonido, Volumen#
```

El volumen es un número flotante y va desde 0 hasta 1. Por lo tanto, un volumen medio sería 0,5.

Además de este parámetro, podemos modificar otros como el panorámico y el tono con las instrucciones "SoundPan" y "SoundPitch".

```
SoundPan Variable del Sonido
Panorámico#
```

LoopSound
PlaySound
SoundVolume
SoundPan
SounPitch

2

Comandos para la utilización básica de sonidos.

El panorámico es un valor flotante y va desde -1 (altavoz izquierdo) hasta 1 (altavoz derecho), pasando por el valor 0 (centro).

```
SoundPitch Variable del Sonido,
Frecuencia en Hercios#
```

Con esta instrucción alteramos el tono del sonido modificando su frecuencia en Hercios. Por ejemplo, calidad CD = 44 Khz, es decir, 44000 Hercios.

Hay que tener en cuenta que para que estos comandos hagan efecto debemos utilizar "PlaySound" para reproducir los sonidos modificados.

CANALES

El uso de canales nos evita tener que utilizar "PlaySound" cada vez que queremos reproducir una modificación de un sonido. En realidad, cuando modificamos un sonido utilizando canales lo estamos haciendo directamente en memoria a través de la variable que manipula el sonido. Un sencillo ejemplo lo tenemos cuando queremos parar la ejecución de un sonido y volver a reproducirlo pasados unos segundos. Para ello, disponemos de las instrucciones "PauseChannel" y

StopChannel
PauseChannel
ResumeChannel
ChannelVolume
ChannelPan
ChannelPitch

3

Comandos para la utilización de canales de audio.

"ResumeChannel" (Ver "ejemplo1.bb"):

```
Global Sonido = PlaySound
(LoadSound ("C:\musica.wav"))
PauseChannel Sonido
Delay 4000
ResumeChannel Sonido
```

Si lo que queremos es parar la reproducción definitivamente debemos utilizar "StopChannel":

```
StopChannel Variable que
manipula el canal de audio
```

Otras opciones muy interesantes que nos permiten la manipulación de canales es la posibilidad de modificar el volumen, tono o panorámico de cada uno de los sonidos en tiempo real (mientras suena) y de forma independiente.

Si queremos modificar el volumen utilizaremos el comando "ChannelVolume". Para el panorámico, "ChannelPan". Y para el tono, "ChannelPitch" (Ver "ejemplo 2.bb").

En ocasiones necesitaremos saber si un canal de audio sigue sonando o no, por ejemplo, para controlar la reproducción de la música de fondo. Para obtener esta información disponemos de la instrucción "ChannelPlaying" (Ver "ejemplo 3.bb"):

```
ChannelPlaying (canal)
```

Esta función devolverá un 1 si el canal de audio sigue sonando y un 0 cuando no.

```
Global Sonido = PlaySound
(LoadSound ("musica.wav"))
While ChannelPlaying(Sonido) = 1
Wend
Print "Se acabó la música"
```

MÚSICAS

Blitz3D tiene un comando que permite cargar y reproducir el audio de una música, "PlayMusic". Admite los siguientes formatos:

RAW, MOD, XM, S3M, MID, RMI, IT, WAV, MP2, MP3, OGG, ASF y WMA.

```
Musica = PlayMusic ("Musica_Menu.mp3")
```

En esta línea de código cargamos la música y la reproducimos una sola vez. Si volvemos a llamar a la función "PlayMusic", el fichero se volverá a cargar y reproducirá de nuevo la música:

```
Musica = PlayMusic
("Musica_Menu.mp3")
While Not KeyHit(1)
 If ChannelPlaying (Musica) = 0
  Musica = PlayMusic
("Musica_Menu.mp3")
 Endif
Wend
```

Como podemos comprobar en este ejemplo, debemos utilizar los comandos para manejar canales si queremos trabajar con ficheros de música. Así, "ChannelPlaying" nos avisa cuando la música ha terminado de sonar, por lo que tenemos que cargarla de nuevo. No es conveniente utilizar "PlayMusic" mientras estamos realizando procesos de pintado de gráficos ya que obtendríamos un gran retardo. Sería conveniente utilizar mejor "LoopSound" y "PlaySound" en su lugar.

EL CD

Además de reproducir ficheros podemos tocar cualquier pista de música procedente de un CD con la instrucción "PlayCDTrack":

```
PlayCDTrack pista [,modo
reproducción]
```

Si en el modo de reproducción ponemos 1, la pista se tocará una vez. Si utilizamos el modo 2, lo hará indefinidamente y si usamos el modo 3 se reproducirá el CD desde la pista elegida hasta el final del CD:

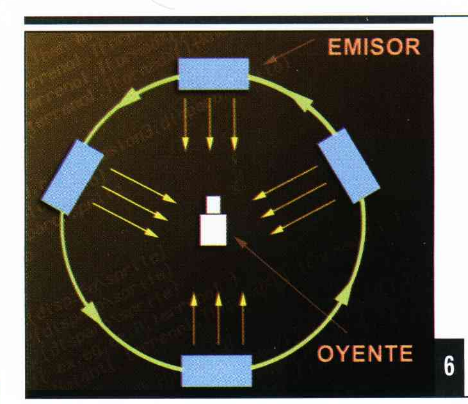
```
Pista = PlayCDTrack (2, 1)
```

SONIDO 3D

Todas las funciones de audio descritas hasta ahora funcionan perfectamente en un entorno 3D. Sin embargo, Blitz3D proporciona la posibilidad de crear y manipular sonido 3D real. Para utilizar este sistema tenemos que tener en cuenta que la reproducción del sonido se hará desde una entidad que esté presente en el entorno 3D, definida como "emisora" del sonido. Para que los sonidos sean escuchados desde el mundo 3D es imprescindible crear un oyente, es decir, una entidad. Así que ésta es la primera operación que tenemos que hacer. Para ello, utilizaremos el comando "CreateListener":

```
CreateListener Entidad oyente
[,factor de proporción]
[, escala del efecto doppler]
[,distancia]
```

La opción "factor de proporción" se refiere a la proporción del volumen del sonido en la distancia. Por defecto vale 1. La "escala del efecto doppler" también vale por defecto1. Y la "distancia" se utiliza para escalar manualmente la distancia que puede oír.



Esquema del funcionamiento del sonido 3D.

```
Sonido = Load3DSound("s.wav")
Micro = CreateListener (Camara)
Emisor = LoadMesh ("emisor.3ds")
...
EmitiSound ( Sonido, Emisor)
...
```

5

Esquema del uso básico de sonido 3D.

```
Camara = CreateCamera()
Oyente = CreateListener (Camara)
```

En este ejemplo asignamos a la cámara como oyente. Generalmente, ésta es la entidad más usada.

A continuación, debemos cargar los sonidos. Para decirle al programa que el sonido que cargamos se utilizará como 3D debemos utilizar "Load3Dsound". Esta instrucción funciona exactamente igual que "LoadSound":

```
Sonido3D = Load3Dsound
("Disparo.wav")
```

Y por último, para reproducir el sonido debemos asignar una entidad emisora, es decir, la entidad que produce el sonido. Por ejemplo, si un OVNI enemigo ha disparado, asignamos esta entidad como emisora del sonido del disparo. Para ello, utilizamos la función "EmitSound":

```
EmitSound (Disparo, Entidad_OVNI)
```

En el siguiente ejemplo, realizamos todos los procesos básicos:

```
Camara = CreateCamera()
Oyente = CreateListener (Camara)
Disparo = Load3Dsound ("Disparo.wav")
OVNI = LoadMesh ("OVNI.b3d")
...
If KeyHit (57) EmitSound
(Disparo, OVNI)
...
```

(Ver "ejemplo 4.bb").

VÍDEO

La carga, reproducción y manipulación de ficheros de vídeo es posible gracias al uso que Blitz3D hace de las DirectShow. Por lo tanto, es necesario tenerlas correctamente instaladas en el sistema. Nuestras posibilidades con estos ficheros se extienden a los formatos GIF, AVI y MPEG. Para cargar cualquiera de estos ficheros disponemos de la instrucción "OpenMovie":

```
Pelicula = OpenMovie
("Introducción.avi")
```

Este comando, además de cargar la película, la reproduce. Sin embargo, solo funciona para los formatos AVI y MPEG. Si queremos reproducir un fichero en GIF debemos utilizar "DrawMovie". Si el fichero no existe no ocurrirá ningún error, simplemente no se reproducirá nada.

Para borrar un fichero de vídeo que ya hemos utilizado tenemos "CloseMovie".

Otra forma de reproducir una película con más control es usando "DrawMovie". Esta función además de ser la única que reproduce ficheros GIF también se puede utilizar en ficheros AVI y MPEG. Funciona dibujando la película en el búfer actual cada frame:

```
DrawMovie Variable de la
película [, PosX, PosY]
[, Ancho, Alto]
```

"PosX" y "PosY" se refieren a la posición de la esquina superior izquierda de la película. Y "Ancho" y "Alto" a su tamaño. Por lo tanto, es posible crear un efecto 16:9 estrechando la película reproducida sobre un fondo negro. Cuando estamos utilizando ficheros en formato GIF, es necesario manejar la animación manualmente, ya que "DrawMovie" solo visualiza el siguiente frame cada vez que es llamado (Ver "ejemplo 5.bb").

En ocasiones, necesitaremos saber el tamaño de la película contenida en el fichero de vídeo. Para obtener esta información disponemos de las funciones

```
Peli = OpenMovie ("peli.wav")
While MoviePlaying (Peli)
  DrawMovie Peli
  Flip
Wend
CloseMovie Peli
```

6

Ejemplo de reproducción de un vídeo en Blitz3D.

"MovieWidth" para saber el ancho y "MovieHeight" para el alto:

```
Pelicula = OpenMovie
("Introducción.avi")
Print "La película mide
"+MovieWidth (Pelicula) +
" x "+MovieHeight(Pelicula)
```

Y para terminar, también necesitaremos saber si una película está o no en reproducción. Para ayudarnos a saberlo podemos utilizar "MoviePlaying", el cual devolverá un 1 si la película no ha terminado y un 0 en caso contrario:

```
presentacion=OpenMovie
("presentacion.mpg")
While MoviePlaying
(presentacion)
 DrawMovie presentacion,0,50,
 800,500
 Flip
Wend
CloseMovie presentacion
```

Hemos podido comprobar que las capacidades multimedia que nos ofrece Blitz3D son realmente sencillas de manejar y con la posibilidad de utilizar los formatos más comunes y versátiles de la actualidad.



NOTA

Para reproducir películas en formato AVI o MPEG utilizamos "OpenMovie" y para reproducir ficheros GIF debemos usar "DrawMovie".

EL PROGRAMA "MULTIPLAY"

Para ilustrar las funciones básicas para el manejo del audio y vídeo, hemos realizado un pequeño ejemplo de reproductor multimedia que admite ficheros de audio y de vídeo. Le hemos llamado "MultiPlay" y básicamente consiste en elegir un método u otro de reproducción dependiendo del fichero cargado. El programa muestra en la parte inferior de la pantalla un sencillo transportador donde el usuario puede reproducir, pausar o parar un fichero de audio, música o vídeo. Para cargar los ficheros utilizamos la función "Carga_Media()". En esta función visualizamos solo los ficheros de sonido y vídeo. Para ello, obtenemos el nombre y la extensión del fichero y aislamos esto último para controlar el tipo de archivo:

```
Dir=ReadDir(CurrentDir$()) :
Locate 0,0
Repeat
  f$=NextFile$(Dir)
    If f$="" Exit
    If FileType(CurrentDir$()+"\"+f$)=2
      Print "Directorio: "+f$
    Else
      E$=Right(f$,4)
      If E$=".wav" Or E$=".mp3" Or
      E$=".ogg" Or E$=".avi" Or
      E$=".mpg" Or E$=".s3m" Print
      "Fichero:"+f$
    End If
Forever
CloseDir Dir
```

Seguidamente, pedimos el fichero a reproducir y, de nuevo, según su extensión lo clasificamos utilizando una estructura "Select.. Case". Según la extensión cargamos en la variable "Media" el fichero elegido con las instrucciones adecuadas (Ver Código 1).

Antes de seguir, hay que recordar que si utilizamos el comando "PlayMusic" necesitamos asignar el archivo a un canal para poder parar o pausar su reproducción con "StopChannel", etc... Una vez almacenado el fichero que vamos a reproducir volvemos al bucle principal.

Para controlar la posición del puntero del ratón para saber qué elección se ha tomado utilizamos una sencilla comparación de coordenadas (Ver Código 2).

Para terminar, utilizamos otra estructura "Select..Case" para controlar las diferentes reproducciones según el tipo de fichero. Para cada tipo disponemos de diferentes acciones, las cuales vienen representadas por un número almacenado en la variable "Accion":

- **Accion = 0** Stop
- **Accion = 1** Play
- **Accion = 3** Pausa y Resume
- **Accion = 4** Volumen

Por ejemplo, para el control del manejo de un fichero de audio (Ver Código 3).

El manejo de los demás tipos de ficheros es idéntico. Evidentemente, la diferencia radica en el uso de las instrucciones adecuadas para cada uno de ellos.

Código 1. Cargamos en "Media" el fichero elegido

```
Extension$=Right(Fichero$,3)
Select Extension$
  Case "wav","mp3","ogg"    Media=LoadSound(Fichero$): Tipo=1
  Case "mid","s3m"          Media=PlayMusic(Fichero$): StopChannel
                            Media: Tipo=2
  Case "avi","mpg"          Media=OpenMovie(Fichero$): Tipo=3
End Select
```

Código 2. Comparamos las coordenadas

```
X=MouseX()
Y=MouseY()
DrawImage puntero,X,Y
If MouseDown(1) And Media<>0
     If (X>360 And X<390) And (Y>375 And Y<425) Accion=1; Play
     If (X>118 And X<165) And (Y>375 And Y<425) Accion=0; Stop
     ............
EndIf
```

Código 3. Tipos de acciones

```
Select Tipo
  Case 1 ; Sonido
    Select Accion
      Case 0   If ChannelPlaying (Sonido) StopChannel (Sonido)
      Case 1,2 If Not ChannelPlaying (Sonido) Sonido=PlaySound (Media)
      Case 3   SW_Pausa=1-SW_Pausa
               If SW_Pausa
                 PauseChannel (Sonido)
               Else
                 ResumeChannel (Sonido)
               EndIf
               Accion=-1
      Case 4   Volumen#=Volumen+(.01*signo)
               If ChannelPlaying (Sonido) ChannelVolume Sonido,
               Volumen#: Accion=-1
    End Select
...
```



En el próximo número...

... empezaremos con las funciones de red y manejo de internet.

Realización de un vídeo para nuestro juego (II)

En el número anterior nos quedamos en la elección inicial del tipo de plantilla de trabajo. Continuamos en esta entrega con el montaje del vídeo.

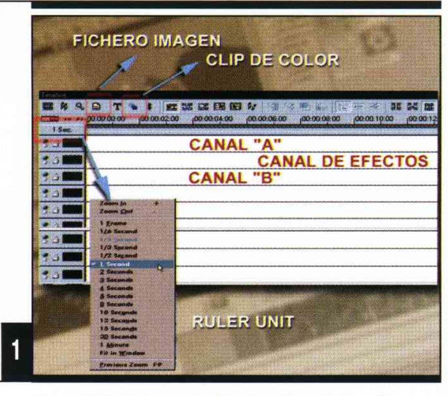
PRIMEROS PASOS

Ulead MediaStudio se caracteriza por ser un programa muy sencillo e intuitivo de manejar. Lo que primero tenemos que saber es que la película se monta en la ventana *Timeline*, la cual se compone de dos partes: las pistas de vídeo y las pistas de audio. En la primera de ellas es donde se monta todo lo relacionado con la imagen: vídeos, dibujos, textos y efectos. La primera pista del *Timeline* es el canal de vídeo "A" y la tercera el canal "B". Ambos canales pueden ser mezclados utilizando multitud de efectos, que se colocan en la pista segunda. Solo las pistas "A" y "B" pueden ser mezcladas con efectos, el resto de pistas pueden contener los mismos elementos pero los efectos de transición no le afectarán. Para nuestro vídeo solo utilizaremos los canales "A" y "B" y el canal de efectos. Además de los efectos típicos de transición, cada clip de vídeo puede llevar sus propios efectos. Estos efectos son de dos tipos: filtros de vídeo y trayectorias de movimiento.

Los filtros de vídeo nos ayudan a añadir multitud de efectos de imagen como desenfoque, cambios de apariencia, de color, etc. Las trayectorias nos permitirán añadir al clip movimientos laterales o cambios de tamaño, entre otras. Por cada clip podemos tener todos los filtros de vídeo que queramos, sin embargo, trayectorias solo podemos tener una a la vez.

Jugando con las demás pistas se pueden obtener efectos de croma ideales para realizar titulaciones.

> **NOTA**
>
> **Llamamos "clip" a cualquier tipo de fichero de vídeo, imagen, texto o fondo de color, así como ficheros de audio.**



Situación de los canales de vídeo "A" y "B. Con la opción Ruler Unit podemos cambiar la regla de la línea de tiempo.

MONTANDO EL VÍDEO DE PRESENTACIÓN

Antes de empezar necesitaremos el siguiente material para realizar nuestro vídeo: Pantallas de los logos del distribuidor, desarrollador y del título del juego, todas las pantallas de texto que desarrollamos en el número anterior y por último, una veintena de capturas del juego. De todas formas, en el directorio "Extras" del CD se encuentra todo este material. Según el guión del vídeo, empezamos mostrando la pantalla del distribuidor lentamente desde el fondo negro (*Fade in* de negro). Para ello, debemos colocar un clip correspondiente a una pantalla en negro de dos segundos de duración. Colocamos la duración de la línea de tiempo en un segundo modificando la "Ruler unit" (Ver Fig. 1).

Insertamos un clip de color del menú *Insert / Color Clip...* o bien pulsando en el icono .



Descripción básica de la ventana de opciones del clip de color.



Descripción básica de la ventana de opciones de los efectos de transición.

4 CAMBIO DE DIRECCIÓN

Procedimiento para añadir un efecto de fundido Crossfade.

Aparecerá una ventana llamada *Colors Options* en donde podemos elegir un color o un degradado para el clip. En la figura 2 se explican las partes de esta ventana de opciones.

Pulsando en el cuadrado de color situado junto a *Color* abrimos el selector de colores y elegimos el negro. El *Keyframer* nos permite seleccionar distintos colores en diferentes puntos en la duración del clip. Para seleccionar el negro como único color del clip pulsamos sobre el botón

NOTA

El funcionamiento del *Keyframer* es el mismo en todas las ventanas de opciones.

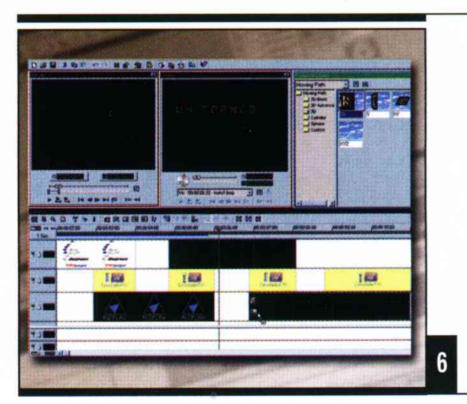


Procedimiento para generar un previo del montaje.

"Pure Color". Al pulsar "Ok" obtendremos una imagen de color negro con un segundo de duración (duración estándar). Colocamos el clip al principio del canal "B" de vídeo. Sobre el extremo derecho hacemos clic y, sin dejar de pulsar, desplazamos el ratón hacia la derecha para alargar el clip hasta los dos segundos. Añadamos a continuación la pantalla del distribuidor. Para obtener una imagen desde un fichero pulsamos en el icono  o bien en *Insert / Image File*. Seleccionamos la imagen "logo_iberprensa.jpg" y la colocamos justo en el tiempo "00:00:01:00" (correspondiente al primer segundo) del canal "A". Vamos a realizar una mezcla entre el clip de color negro (canal "B") y la imagen del logo (canal "A"). Por lo tanto, debemos colocar en el canal de efectos un efecto de transición *Crossfade*. Nos dirigimos a la ventana *Production Library* y abrimos en la librería *Transition Effects* la carpeta *F/X*. Hacemos clic sobre el efecto *Crossfade* y sin dejar de pulsar lo llevamos con el ratón hacia el canal de efectos (pista 2). Al soltarlo, se abrirá la ventana de opciones del efecto elegido. En la figura 3 se explican los parámetros de esta ventana.

Hacemos lo mismo para mostrar la pantalla del desarrollador ("logo_M2PLAY.jpg"). La cargamos y la situamos en el canal "B" y colocamos un *Crossfade* entre los dos logos (Ver Fig. 4).

Antes de seguir, hacemos un inciso para comprobar el resultado de lo que llevamos hecho hasta ahora. Si pulsamos sobre cualquier clip se mostrará el contenido en la ventana *Source*, por otro lado, el resultado lo veremos como previo en la ventana *Preview*. Ulead MediStudio no trabaja en tiempo real, así que tenemos que esperar a



Los clips de vídeo pueden llevar filtros de vídeo y trayectorias de movimiento.

que genere el previo del montaje. Para realizarlo colocamos el ratón sobre la franja situada debajo de la línea de tiempo y lo desplazamos hasta seleccionar lo que queremos visualizar como se muestra en la figura 5.

Para crear el previo pulsamos la tecla "Enter".

El siguiente paso en el guión es empezar a visualizar las pantallas de texto. Realizamos un fundido a negro del logo del desarrollador mezclándolo con un clip de color negro de 3 segundos de duración. Cargamos la primera pantalla de texto "texto1.jpg" y la situamos debajo en el canal "B" para mezclarlo (Ver Fig. 6).

La idea es que mientras aparece el texto desde negro desenfocado se va desplazando hacia el centro de la pantalla a la vez que se enfoca. Luego desaparecerá, desenfocando de nuevo, fundiéndose con el siguiente texto. Para realizar estos procesos tenemos que aplicar al clip un filtro de vídeo y una trayectoria con movimiento. Pero dejaremos este proceso para la próxima entrega.

En el próximo número...

... terminaremos el montaje de nuestro vídeo de presentación.

Simuladores deportivos

Siguiendo con nuestra historia de los juegos de simulación para PC, pasamos revista al género deportivo. Básicamente, el deporte siempre ha sido considerado como un juego en sí, sobre todo para las personas que disfrutan viéndolo.

Los amantes del deporte siempre han tenido en mente poder participar con su equipo favorito o sentirse en la piel del deportista de elite del momento. Aprovechando esta premisa, los juegos deportivos para ordenador han estado presentes a lo largo de la historia lúdica para ofrecer una alternativa diferente al entretenimiento. Se puede decir que existen juegos que simulan casi todos los deportes más populares. Por ello, vamos a exponer un breve repaso en la simulación de los siguientes deportes: fútbol, baloncesto, atletismo, tenis, golf, hockey, rugby, boxeo, bolos, billar, etc.

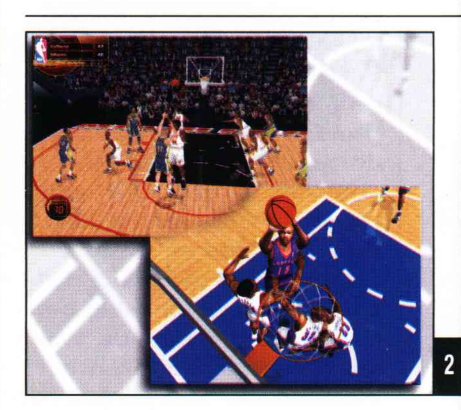
HISTORIA Y EVOLUCIÓN

Empezamos haciendo un breve repaso de la historia del que quizás sea el deporte más presente en los juegos para ordenador, el fútbol. El resto de deportes como el baloncesto, rugby o hockey toman prácticamente el diseño del fútbol. Para comentar la evolución de los juegos de fútbol, debemos mirar atrás, hacia los 8 bits. Fue en el año 1984 cuando se inició realmente este deporte en ordenadores Spectrum y Amstrad con *Match Day* de Jon Ritman (Ocean). El juego tenía una representación lateral y en su primera versión carecía de cualquier elemento estadístico. En realidad se trataba de un arcade en el que lo único era jugar un partido. Aparecieron dos nuevas entregas entre 1987 y 1990 de este título difícil de superar en aquella época. Por entonces, el deporte era el rey en las recreativas y en 1985 aparece un nuevo concepto gráfico con el juego *Tehkan World Cup* (Tehkan), la vista cenital. Esta técnica, unida a los sistemas de scroll, pronto fue adoptada por la mayoría de títulos para ordenador y consola. Aparte del aspecto gráfico, la jugabilidad ha sido siempre el gran reto para este tipo de juegos. Siguiendo esta idea aparece un título desarrollado por Dino Dini para recreativas que marcaría un antes y un después: *Kick Off* (1988, Anco Software). *Kick Off* ofrecía un nuevo concepto en el control del balón en el que adoptaba cierta libertad de movimientos con respecto al jugador. También Dino Dini con *Player Manager* (1990), para Amiga y Atari, abrió las puertas a la posibilidad de mezclar la acción con la gestión de jugadores. Por entonces, el PC estaba en pañales y sus juegos de fútbol eran versiones de ordenadores de 8 bits como *Match Day*, *Emilio Butragueño Fútbol* (1988, Topo Soft) o *Michel Fútbol Master* (Dinamic). Aunque la vista cenital dominaba todavía estos juegos, en las recreativas se empezó a inclinar la cámara para obtener una perspectiva más lateral del campo y los jugadores. Después, se impone la perspectiva isométrica presentada por primera vez en *FIFA International* (Electronics Arts, 1993). La implantación de los sistemas de almacenamiento como CD o disco duro permite que, aparte de una evolución gráfica evidente, el concepto de arcade se una al preámbulo del juego con mayores y mejores sistemas de estadísticas y configuración de equipos proporcionando un

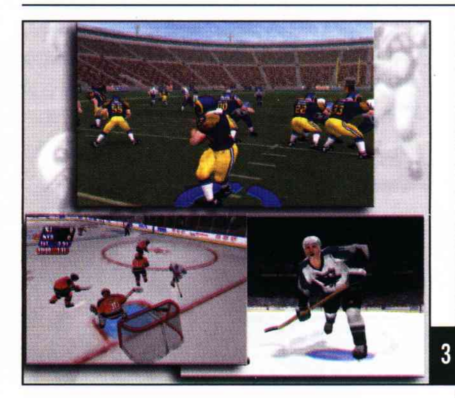


1

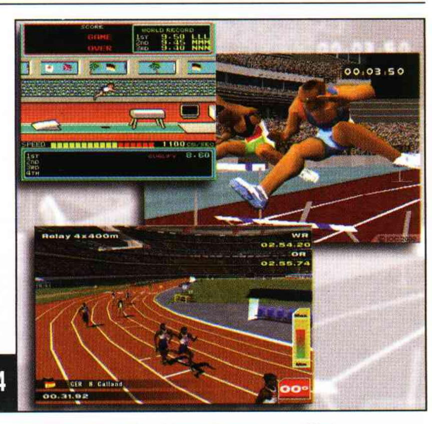
La serie FIFA de Electronics Arts es el simulador de fútbol más popular del mundo del PC.


2

(Arriba) La incursión de Microsoft en los simuladores de baloncesto con NBA Drive 2000. (Abajo) La serie NBA Live de EA Sports.


3

(Arriba) Madden NFL, un espectacular ejemplo de simulador de Rugby. (Abajo) La serie NHL de simuladores de Hockey también de EA Sports.



4

(Arriba Izquierda) HyperSports, un clásico entre los juegos de atletismo. (Arriba Derecha) Sydney 2000 fue un título con escasa calidad gráfica pero con excelente cinemática. (Abajo) La apuesta de Dinamic Multimedia por el atletismo con PC Atletismo 2000.

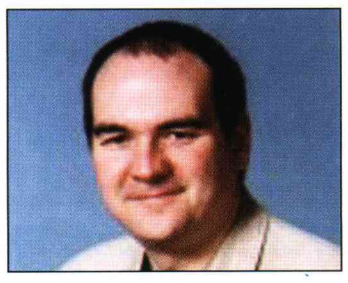


LA BIOGRAFÍA...

DINO DINI

Creador de **Kick Off**

Dino Dini entró en la historia de los videojuegos después de escribir para Anco Software en 1988 el juego de fútbol *Kick Off*. Con este título, Dino marcó una forma de simular este tipo de deportes en un juego redefiniendo el fútbol de acción. Su vida ronda en torno al desarrollo de juegos de fútbol y es el responsable del desarrollo y diseño de todos los títulos para diferentes plataformas. Sus juegos han sido ampliamente galardonados desde 1989. Con más de 20 años de experiencia a la espalda funda en el 2001 su propia desarrolladora llamada Abundant Software afincada en Bristol (UK) y principalmente dedicada a los juegos de fútbol.



Dino Dini.

mayor equilibrio entre la simulación y la jugabilidad. No podemos obviar la excelencia en este sentido dada por *International SuperStar Soccer* (Major A, Konami) para consola y (padre del actual *Pro Evolution Soccer*) posteriormente en compatibles. O la serie *PC Fútbol* (93-2001, Dinamic Multimedia) y *Champion Ship Manager* (1999, Sports Interactive). La vista lateral con scroll era el rey en el mercado hasta que aparecen las posibilidades 3D con las DirectX. Las sagas más populares se acogen a este nuevo sistema de visualización. Podemos hablar de un resurgir en los juegos de fútbol y simuladores deportivos en general. Es la casa Electronics Arts (EA) la que innova en este aspecto con *FIFA 96*, en la que se representa en 3D el entorno y también los jugadores en *FIFA 97*. Hasta la fecha la serie *FIFA* licenciada por EA ha acaparado toda la atención por los jugadores de todo el mundo y es con *FIFA 2003* donde el fútbol adquiere su mayor exponente. Actualmente los juegos deportivos y en particular los de fútbol presentan todos los avances tecnológicos para ofrecer una simulación cada vez más real del juego. Por ejemplo encontramos: jugadores y equipos reales, gráficos 3D, varias cámaras, repetición de jugadas, comentaristas en tiempo real, climatología, interfaz de juego inteligente, sistema de movimientos para jugadores por *motion capture*, física realista del balón, etc.

Al igual que el fútbol, otros deportes como el baloncesto siguieron la misma trayectoria técnica, aunque con una menor complicación debido al menor número de jugadores y al tamaño del terreno de juego. La mayoría de títulos para PC dedicados al baloncesto se centran en la liga NBA Estadounidense y de nuevo EA empieza a explotar el filón que supone esta licencia publicando las series *NBA Live* (95, 96 y 97). A partir del año 99 los conceptos básicos se centran en la espectacularidad visual (sobras dinámicas, reflejos, motion capture) que brindan las nuevas tarjetas gráficas y en las posibilidades de combinación de jugadas. Los más destacables aparecen a partir de 1999 con *NBA Drive 2000* (Microsoft), o la serie *NBA Live* (2000- 2002, EA Sports) o su rival más directo *NBA Basketball 2000*.

Debemos hacer una mención especial a Electronics Arts por la creación del sello EA Sports para la comercialización de todos los deportes más significativos del mundo. Así, aparte de la serie *FIFA* de fútbol o *NBA* de baloncesto, encontramos otros deportes como el rugby o el hockey bien representados por la serie *Madden NFL* y *NHL* respectivamente. Quizás la serie de fútbol americano *Madden NFL* supera con creces a la serie *FIFA* en calidad técnica, claramente expuesta en *Madden NFL 2001*. Las animaciones de cada jugador son totalmente realistas al igual que la IA de equipo.

Por otro lado, la simulación del atletismo no ha estado muy explotada en el sector lúdico. Prácticamente, solo aparecen nuevos títulos acompañando cada edición de los Juegos Olímpicos, es decir, cada cuatro años. Desde aquellos primeros títulos para 8 bits como *HyperSporst* (Imagine) o *Daley Thompson´s Supertest* (Ocean) en donde hundíamos las teclas del teclado para que nuestro deportista corriese más y más hasta los modernos sistemas de control mediante el ratón de *PC Atletismo 2000* (Dinamic Multimedia) con 19 pruebas deportivas o *Sydney 2000* (Eidos) con 12.

En el próximo número...

... terminaremos de ver el resto de los deportes representados por un juego de ordenador como el golf, el tenis y los deportes de invierno. Además entraremos de lleno en los simuladores de velocidad.

Cuestionario Videojuegos 17

Preguntas

1. **En Blitz3D escribe un programa para cambiar el volumen simultáneamente a dos sonidos con los cursores del teclado.**

2. **En Blitz3D escribe el código para poder visualizar en pantalla un mosaico de cuatro pantallas de un vídeo de resolución 800 x 600.**

3. **Escribe el código necesario para desplazar un objeto 3D sobre un terreno hacia una posición cambiante.**

4. **¿Qué clase de comportamiento podemos implementar para un grupo de entidades que luchan entre sí en una zona de juego?**

5. **¿Qué diferencia hay entre la primera textura y la segunda en un proceso de *Blending* para un terreno?**

6. **¿Qué tenemos que tener en cuenta en una textura utilizada en un procedimiento de tileado?**

7. **Procedimientos para crear un loop en Anvil Studio.**

8. **¿Qué es la opción *Performer* de Anvil Studio?**

9. **¿Cuántas pistas de vídeo podemos mezclar con efectos de transición en Ulead MediaStudio?**

10. **¿Cómo podemos aplicar un efecto de desenfoque a un clip de imagen en MediaStudio?**

Respuestas al cuestionario 16

▷ **1.**

```
Directorio=ReadDir("c:\")
Repeat
  Fichero$=NextFile(Directorio)
  Print Fichero$
Until Fichero$=""
```

▷ **2.**

```
Fichero = WriteFile("datos.dat")
For n = 1 To 10
Numero=Rand (1,100)
WriteInt ( Fichero, Numero )
Next
CloseFile (Fichero)
```

▷ **3.** Una entidad puede detectar la proximidad de otra por medio del comando EntityDistance (Entidad_Origen, Entidad_Destino)

▷ **4.** Para que cada entidad de la estructura tenga un comportamiento independiente utilizando la misma función de control.

▷ **5.** Mediante el manejo de un *mesh* con la forma del terreno creado en una aplicación externa de modelado o creando el terreno con Blitz3D desde programación a partir de un mapa de alturas.

▷ **6.** Podemos utilizar un visualizador de terrenos construido en Blitz3D o bien utilizando Bryce, utilizando el método de asignación de imágenes a un terreno en el editor de terrenos.

▷ **7.** Podemos importar directamente un fichero en formato .WAV en una pista de audio a través de la opción *File / Import Audio to Active Track from... / Import entire file.*

▷ **8.** En la sección *Compose* pulsamos en el botón "Add sounds..." Elegimos la pestaña *Audio Samples* y luego en *Edit Samples*. Asignamos la nueva muestra pulsando en el botón "Add New Sample" y elegimos *Import*.

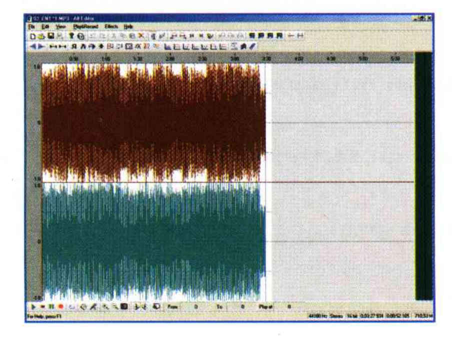
▷ **9.** Podemos obtener capturas de nuestro juego salvando el búfer de pantalla en un fichero de imagen BMP con el comando *SaveBuffer*.

▷ **10.** Lo primero que tenemos que hacer antes de editar con Ulead MediaStudio es crear una plantilla de trabajo.

Contenido CD-ROM 17

▶ AUDIO

■ All Editor 2.2.4



Crea un estudio de música realmente profesional en tu Escritorio.

■ Advanced Audio Corrector

Gracias a esta práctica utilidad, elimina las distorsiones en tus archivos de música.

■ Ogg Encoder Decoder

Crea ficheros Ogg Vorbis (ogg) a partir de MP3 o de WAV.

■ Renoise

Estupendo programa muy profesional con el que podrás componer y editar tus propios ficheros de audio.

▶ DISEÑO 2D

■ Ahaview

Rápido visualizador para saber exactamente dónde están tus imágenes y tener su vista previa.

■ Cam2pc

Este otro visualizador además te permitirá localizar fotos y bajarlas desde internet.

■ Adobe Photoshop Elements 2.0



Demo del sensacional programa de Adobe específicamente diseñado para tratar imágenes digitales.

■ Fineview

Visualizador con el que podrás además realizar ajustes rápidos de brillo, contraste, etc...

■ ImageN

Programa de simpática interfaz con el que podrás manipular tus imágenes, editarlas, compartirlas, etc...

▶ DISEÑO 3D

■ Design Workshop Lite



Estupendo programa de creación 3D de paisajes, entornos y todo lo que se te ocurra.

■ Insta 3D 2.6

Aplicación para que crees tus propios modelos y escenas en tres dimensiones.

■ Mystica 3.6

Utilidad para renderizar imágenes en pocos minutos.

■ People Putty

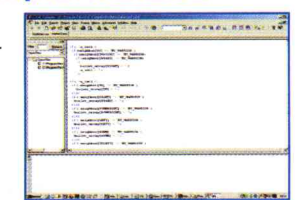
Divertidísimo software para caracterizar personajes con sus correspondientes expresiones.

▶ PROGRAMACIÓN

■ CodeConter Pro 1.0

Cuenta automáticamente las líneas de código, los comentarios, o cualquier parte del programa que desees.

■ CxC Compiler Standard Edition



Potente compilador que sin duda te agradará.

■ Microsoft Java VM

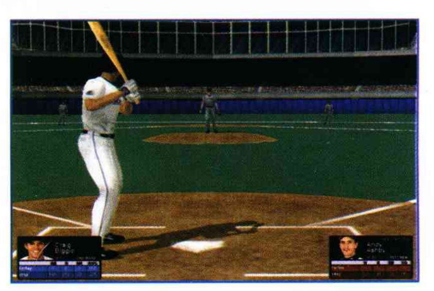
Máquina virtual de Java, muy útil ya que éste se está convirtiendo en un lenguaje cada vez más utilizado.

■ Manhattan 1.0

Controla tus proyectos y su código tanto en C como en Java con esta original aplicación.

▶ JUEGOS

■ Microsoft Baseball 2000



Este estupendo juego contiene todos los equipos, jugadores y estadios oficiales.

■ FIFA 2002

Demo del juego de fútbol más famoso. Excelentes gráficos y un gran realismo.

■ Madden NFL 2003

Juega tu propia versión de la superbowl con este estupendo programa.

■ NBA Full Court Press

Si lo tuyo es el baloncesto, disfrutarás de lo lindo son este simulador.

■ Sydney 2000

Siéntete como un auténtico deportista olímpico y gana medallas compitiendo.

■ Zone of fighters

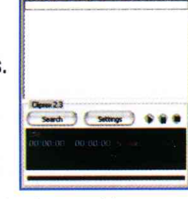
Nueva versión de nuestro juego, con más mejoras.

▶ VÍDEO

■ Bsplayer 0.8

Reproduce rápidamente tus ficheros de vídeo y audio con esta pequeña aplicación.

■ Cliprex Lite 2.3



Otro reproductor de vídeo, que admite además múltiples formatos.

■ DirectX

Últimas versiones para Windows 95, 98, Millenium y 2000, para que aumentes la potencia de tus vídeos.

▶ EXTRAS

En este apartado encontrarás todos los ejemplos de los que hablamos en el coleccionable, para que no pierdas detalle.